REMARKS

Summary of Office Action

Claims 1-13, 15-27, and 29-41 are pending in the present application. Claims 1, 13, 15, 16, 29 and 30 are independent claims.

In the Official Action, dated July 14, 2006, the Examiner finally rejected all pending claims.

The Examiner rejected claims 1, 13, 15, 16, 29 and 30 under 35 U.S.C. § 102(b) as anticipated by User Interface Markup Language Draft Specification, dated January 17, 2000, Copyright Harmonia, Inc., Language Versigon 2.0a ("UIML").

The Examiner rejected claims 2-7, 17-22, 31-36 under 35 U.S.C. § 103(a) over UIML in view of U.S. Patent Publication No. 2003/0070158 ("Lucas").

The Examiner rejected claims 8-9, 23-24, 37 and 38 35 U.S.C. § 103(a) over UIML in view of U.S. Patent Publication No. 2003/0212904 ("Randle").

The Examiner rejected claims 10, 25 and 39 under 35 U.S.C. § 103(a) over UIML in view of Lucas and further in view of Randle.

The Examiner rejected claims 11, 26 and 40 rejected under 35 U.S.C. § 103(a) over UIML in view of U.S. Patent Publication No. 2003/0058277 ("Bowman-Amuah").

The examiner rejected claims 12, 27 and 41 rejected under 35 U.S.C. § 103(a) over UIML in view of U.S. Patent Publication No. 2004/0093344 ("Berger").

Claims 1, 3, 5, 6, 9, 17, 18, 30, 34, 35, 38, and 41 have been amended. Consideration of the amendment and reconsideration of the final rejection is requested.

Summary of Disclosure

The present disclosure is directed to a Type Description Language (TDL). The TDL is an extensible markup language (XML) based language that provides an interface description for mapping an interface specification to its wire format in a deterministic manner. The methodology enabled by the TDL represents the behavioral and data aspects of a type by creating a one to one mapping (injective function) from an abstract type to a schema. The TDL utilizes XML Schemas (XSD) enhanced to represent type references and arrays and numerous syntactic restrictions such as usage of element representation for fields as the canonical syntax to represent the data aspect of a type.

The TDL leverages the duality between the type-based (objects) and XML-based views and may be used for exchanging metadata between various kinds of type (object) systems, such as Component Object Model (COM), Common Object Request Broker Architecture (CORBA), Common Language Runtime (CLR), etc.

Concepts

In analyzing the examiner's rejections and the art cited, it is important to have a clear understanding of a number of concepts embodied in the application and in the art.

I **User Interface**

A User Interface (UI) refers to all of the methods through which a person (user) interacts with a device, system, or computer program. The UI provides the user with a way to provide input to the system and allow the system to provide an output. In computer science UI refers to the graphical, alphanumeric, and auditory information the program presents to the user, as well ask control sequences that the user may employ to control the program.

Application Interface

An Application Programming Interface (API) is an interface that a computer system, library or application (Application) provides to allow requests for services from other computer programs to be processed by the Application. The objective of an API is to enable software developers to access the functions or libraries of an Application without requiring access to the source code of the Application.

Services

A service is a software system designed to support inter operable machine to machine interaction over a network. The service will typically have an interface described in a machine processable format. Systems interact with a service in the mode specified by its interface using messages. Typically the message would be conveyed in XML and may be enclosed in a SOAP envelope. Applications written in different languages and operating on different platforms may use Web services to exchange data over computer networks.

XML Schema

An XML Schema is a description of an XML document expressed in terms of constraints on the structure and content of the document. Schemas provide the sets of rules that define structure, content and semantic of XML documents. An XML Schema is a replacement, more complete of doctype. An XML schema provides a view of the document type at a relatively high level of abstraction. The mechanism for associating an XML document with a schema varies according to the schema language. The association may be achieved via markup within the XML document itself, or via some external means. The process of checking to see if an XML document conforms to a schema is called validation. Documents are only considered valid if they satisfy the requirements of the schema with which they have been associated.

One to One Mapping

A One to One Mapping (injective function) refers to a mapping that connects the members of a set A with members of another set B in a way that a single element of B is associated with each element of A, and no two elements of A map onto the same element of B.

THE CITED REFERENCES

I **User Interface Markup Language (UIML) Draft Specification, document version 17 January 2000 (UIML)**

The objective of UIML is to create an open standard user interface description language in XML that may be implemented by programmers. The idea is to provide tools for the creation of user interfaces on any platform. This object is set out in detail at http://www.uiml.org/as follows:

> "UIML was the result of taking a clean sheet of paper and designing a new language for user interfaces. UIML is an XML-compliant language, so it looks a lot like HTML. UIML is designed to serve as a single language which permits creation of user interfaces for any device, any target language (e.g., Java, C, WML), and any operating system on the device. However, UIML is not a silver bullet: The UI designer must still design separate UIs for each device, and then represent those designs in UIML. UIML does not magically create multiple UIs from a single description; instead it is a language in which those multiple UIs can be recorded."

Thus, UIML does not create user interfaces, but rather is a way of recording multiple user interfaces. Two other observations about the UIML document are important. There is no mention of XML Schemas, and there is no mention of one to one mapping of types to a schema.

Furthermore, the language disclosed in the UIML document requires additional definitions to generate a User Interface. The UMIL document provides:

> "Thus a UIML author needs more than this document, which specifies the UIML language. You also need one document for each UI toolkit (e.g., Java Swing, Microsoft Foundation Classes, WML) to which you wish to map UIML. The toolkit-specific document enumerates for a particular toolkit a vocabulary of toolkit components (to which each part element in a UIML

document is mapped) and their property names." (UMIL, p.10, par. 1).

Lucas (US 2003/0070158, Apr. 10, 2003)

Lucas is directed to programming language extensions for processing data representation language objects and related applications. Lucas provides:

> "For the illustrated embodiment, interpreter/compiler 104 includes an application programming interface (API) (not shown), through which programming statements formed using language extensions of the present invention may be programmatically submitted for compilation by a variety of application-specific processes. For example, in accordance with one embodiment of the present invention, a web server application makes calls to mapping services 100 upon the receipt of XML documents in order to map XML document objects as e.g., internal Java classes for additional processing by the web server. Such application-specific processes may be co-resident with mapping services 100 on the same "host" system (not shown) as mapping services 100, or located remote from the "host" system and communicate with mapping services 100 using conventional cross system communication techniques." (Par. 0024)."

Lucas provides language extensions in mapping objects from a data representation language such as XML to corresponding objects of a programming language such as JAVA. Lucas states:

> As alluded to above, the language extensions of the present invention are particularly well-suited for use in mapping objects from a data representation language [e.g. XML] to corresponding objects of a programming language [e.g. JAVA], and vice versa. Such a language mapping may be desirable in situations where, for example, a system having an internal operating environment based upon a programming language such as Java, is required to

> exchange data with other systems using a data representation language such as XML. (Par. 0048)

Lucas does not mention, disclose, refers to, or implies a methodology including creating a one to one mapping of each type in a device or object to an XML schema. Indeed, there is no mention of one to one mapping or injective functions in the Lucas disclosure.

Randle (U.S. 2003/0212904, Nov. 13, 2003)

Randle discloses a system and method of providing standardized transmission of data by translating non-native requests and responses to and from a normalized format or to a format needed processing the request or response (Abstract, line 1). The disclosure is directed to an enterprise business integration engine having secure access from using the financial industry.

Randle provides:

> "In the preferred embodiment, the adapter 14 comprises software capable of translating and standardizing a semantic, data format, transport, and/or wire protocol of an input signal such as the channel message 20 and communicating the message 20 to a variety of destinations 13a-n. The adapter 14 of the preferred embodiment further uses an XML format to encode the message 20. The adapter 14 then transmits the translated message 24 to the processor 25. Before processing the content of the translated message 24, the sign-on information is validated and a session cache established similar to that described above for the embodiment not requiring the adapter." (Par 59)

Rendell does not disclose a methodology for creating a one to one mapping between each type in a device or object to an XML schema. Indeed there is no mention of an XML schema in the Rendell reference.

Bowman-Amuah (U.S. 2003/005827, Mar. 27, 2003)

Bowman-Amuah is directed to assigning a view to an activity. Notification is received that a startup event of an activity has occurred. A reference to a first instance of an object created by the startup event of the activity is also received. A view to launch is

determined in response to the receipt of the notification and the reference. The view is based on predetermined criteria. The view is associated with the activity and displayed. Bowman-Amuah was cited by the examiner for its reference to peer-to-peer messaging. However, Bowman-Amuah does not make reference to XML Schemas or the one to one mapping of objects types to XML Schemas.

Berger (U.S. 2004/0093344, May 13, 2004)

Berger is directed to a method for mapping enterprise data assets to a semantic information model. Berger describes the method as follows:

> "The present invention provides a method and system for deriving transformations for transforming data from one schema to another. The present invention describes a general method and system for transforming data confirming with an input, or source data schema into an output, or target data schema." (Par. 47)

Thus, Berger is concerned with mapping a schema to a schema. Berger does not disclose a methodology that includes a one to one mapping of each type in a device or object to XML schema.

The Claims

I **Claim 1**

Claim 1 has been amended clearly set out the one to one mapping character of applicant's methodology. Of similarly, claim one has been amended by deleting the reference to "vice versa." That redundancy is unnecessary in view of the meaning of a one to one mapping which is an element of the claim. Similar amendments have been made to the other claims to remove this redundancy. The examiner rejected claim 1 under 35 USC section 102 (b) as being anticipated by UIML. Although the examiner asserted that UIML discloses a one to one mapping each type of a particular type based system to an XML schema there is no such disclosure in UIML. As stated above in the discussion regarding this reference, UIML does not mention one to one mapping and does not mention XML Schemas.

The structure of a UIML document is described in section 3.1 (page 9) as follows:

> In UIML version 2.0a, a UI is a set of interface elements with which the end user interacts. Each interface element is called a part; just as an automobile or a computer is composed of a variety of parts, so is a UI. The parts may be organized differently for different categories of end users and different families of devices. Each interface part has content (e.g., text, sounds, images) used to communicate information to the end user. Interface parts can also receive information from the end user using interface artifacts (e.g., a scrollable selection list) from the underlying device. Since the artifacts vary from device to device, the actual mapping (rendering) between an interface part and the associated artifact (widget) is done using a style element.

The preceding description of the structure of a UIML document is different from the interface description described by applicant suggests a one too many mapping between interface parts and end user interface artifacts.

It is respectfully submitted that UIML does not anticipate claim 1. Reconsideration and allowance is requested.

Claim 2

The examiner rejected claim 2 under 35 USC section 103 (a) as obvious over UIML in view of Lucas. Claim 2 is the original claims submitted in the application and depends from amended claim 1 and fertile defines the XML based interface description language as being a type description language (TDL). As stated above, neither UIML nor Lucas disclose a methodology including the creation of a one to one mapping of each type in a device or object to an XML schema. Furthermore neither UIML nor Lucas discloses the use of an XML based interface description language or a type description language. Lucas is directed to a system for manipulating data representation language based objects in a native programming language environment. Not only does Lucas not describe a one to one mapping, looking closely at the claims, it is clear that Lucas contemplates a one to many mapping. Claim 17 of Lucas provides:

> "17. A method comprising: receiving one or more definitional statements expressed with language elements of a programming language; determining whether the one or more definitional statements includes a statement designed to import a predefined data type description describing a class of objects associated with said data representation language; **determining whether the one or more definitional statements associate one or more data representation language values with said data type description**; determining whether the one or more definitional statements include an operator, wherein said one or more data representation language values are operands of said operator; determining whether said operator will result in one or more of said data representation language values that do not conform with constraints specified by said data type description; and generating one or more error messages identifying said operator as generating results that do not conform with constraints specified by said data type." (Emphasis added).

It is respectfully submitted that claim 2 is patentable over UIML in view of Lucas. Reconsideration and allowance is requested.

Claim 3

Claim 3 has been amended to conform with the amendment in claim 1. The examiner rejected claim 3 under 103 (a) over UIML in view of Lucas. The basis for the rejection was that in the view of the examiner UIML discloses a one to one mapping from a programming construct to an XML Schema. In applicant's view, UIML does not disclose such a one to one mapping. In this regard, the language from UIML cited by the examiner is illuminating. UIML provides:

> As discussed earlier, UIML captures the elements that are common to any UI: an enumeration of the UI parts, events that occur for those parts, presentation style, content, and interconnection to application logic. The UIML author specifies instance and class

> names of their own choice for interface parts, events, and methods. These names are mnemonics for the interface implementer. The UIML document specifies a mapping from those names to names that are vocabulary specific to a particular target platform. For example, if the target is Java AWT, the vocabulary might consist of the "java.awt." and "java.awt.event." class names, such as "Frame," "Menu," and "Button." If the target is WML, the vocabulary might be tag names, such as "card," "select," and "input." **The vocabulary of target platforms is not a part of UIML.** That vocabulary only appears in UIML as the value of attributes in UIML. **Thus UIML only needs to be standardized once, and different constituencies of end users can define vocabularies that are suitable for various toolkits independently of UIML.** In addition to creating vocabularies for particular toolkits (e.g., Java AWT), a vocabulary for generic classes of toolkits (e.g., mapping to any graphical UI) could be devised. Or new vocabularies can be defined as new devices and UI technologies are created in the future. (Emphasis added)

Thus the functionality of the UIML methodology does not provide the deterministic mapping that results from a one to one mapping is required by applicant's claims. It is respectfully submitted that claim 3 is patentable over UIML in view of Lucas. Reconsideration and allowance is requested.

Claim 4

The examiner rejected claim 5 under 35 USC section 103 (a) over UIML in view of Lucas. The examiner stated that the code on page 14-15 all of UIML discloses that the programming construct is a class programming construct. Stating it doesn't make it so. Again, UIML fails to disclose the need for a one to one mapping that is recited in claim 1 on which claim 4 depends. The cold referenced by the examiner does not show the creation of a one to one mapping from a programming construct to an XML schema for describing the programming construct.

It is respectfully submitted that claim 4 is patentable over UIML in view of Lucas. Reconsideration and allowance is requested.

Claim 5

The examiner rejected claim 5 under 35 USC section 103 (a) over UIML in view of Lucas. The basis of the rejection was the same as for claim 4. It is respectfully submitted that claim 5 is patentable for the same reasons as set forth for claim 4 above. Reconsideration and allowance is requested.

Claim 6

The examiner rejected claim 6 under 35 USC section 103 (a) over UIML in view of Lucas. The examiner cited the same language in UIML for the rejection of claim 6 as for the rejection of claim 4. Applicant submits that UIML does not disclose the one to one mapping recited in the claim.

It is respectfully submitted that claim 6 is patentable for the same reasons as set forth for claim 4 above. Reconsideration and allowance is requested.

Claim 7

The examiner rejected claim 7 under 35 USC section 103 (a) over UIML in view of Lucas. Section 8.3 of the UIML reference disclosed those the organization all JAVA AWT classes in a hierarchy in which each child class inherits the parent class attributes. Claim 7 of the press an application is directed to ATP held that supports inheritance of programming constructs in a method for providing an interface description involving an element of creating a one to one mapping each type of the construct to an XML schema. Is respectfully submitted that the language in section 8.3 of UIML does not disclose the claim language.

It is respectfully submitted that claim 7 is patentable over the cited art for those reasons and for the reasons set forth for claim 4 above. Reconsideration and allowance is requested.

Claim 8

The examiner rejected claim 8 under 35 USC section 103 (a) over UIML in view of Randle. The following language from Randall his argue to disclose that the XML based IDL is a wire format for message communications relating to the service between devices of the

computing system, in a method of providing an interface description including creating a one to one mapping of each type in the devised for object to an XML schema.

> "In the preferred embodiment, the adapter 14 comprises software capable of translating and standardizing a semantic, data format, transport, and/or wire protocol of an input signal such as the channel message 20 and communicating the message 20 to a variety of destinations 13a-n. The adapter 14 of the preferred embodiment further uses an XML format to encode the message 20. The adapter 14 then transmits the translated message 24 to the processor 25. Before processing the content of the translated message 24, the sign-on information is validated and a session cache established similar to that described above for the embodiment not requiring the adapter."

There is no reference to an XML schema in the Randle publication. There is also no reference in the Randle publication to a one to one mapping as described in claim 8. Randle is directed to a system on method for providing standardized transmission of data, and not for providing an interface description of the service. The combination of UIML and Randle failed to disclose the creation of a one to one mapping of each type in the device or object to an XML schema.

It is respectfully submitted that claim 7 is patentable over the cited art. Reconsideration and allowance is requested.

Claim 9

The examiner rejected claim 9 under 35 USC section 103 (a) over UIML in view of Randle. The examiner argued that it would have been obvious to one of ordinary skill in the art, having the teachings of UIML and Randle to modify UIML to include mappings between wire format and communications as taught by Randle. Claim 9, as amended, recites the element of creating a one to one mapping from the wire format to the message communications. Such one to one mapping is not disclosed either by UIML and nor Randle. The combination of the reference does not suggest the element of creating a one to one mapping from the wire format to the message communications.

It is respectfully submitted that claim 9 is patentable over the cited art. Reconsideration and allowance is requested.

Claim 10

The examiner rejected claim 10 under 35 USC section 103 (a) over UIML in view of Lucas and further in view of Randle. Claim and is directed to the transfer of a service reference across an application boundary. Randle is directed to providing a security integrator capable of aggregating and consolidating authentication and/or authorization protocols performed by multiple related or indirectly related channels and systems. Randle does not suggest the transfer of this service reference across an application boundary. Additionally, as stated above, neither UIML and/or Lucas disclose the one to one mapping element of the claim.

It is respectfully submitted that claim 10 is patentable over the cited art. Reconsideration and allowance is requested.

Claim 11

The examiner rejected claim 11 under 35 USC section 103 (a) over UIML in view of Bowman-Amuah. Bowman-Amuah was cited for the mention of a peer-to-peer system. Neither UIML nor Bowman-Amuah disclose the creation of a one to one mapping of each type in the devised or object to a Lanham's schema. Neither reference discloses a method for providing an interface description a peer-to-peer computing system by creating such a one to one mapping.

It is respectfully submitted that claim 11 is patentable over the cited art. Reconsideration and allowance is requested.

Claim 12

The examiner rejected claim 12 under 35 USC section 103 (a) over UIML in view of Berger. Claim 12 recites a method of providing an interface description including creating a one to one mapping for each type in a device or object to an XML schema, describing the one to one mapping using an extendable markup language based interface description language, and mapping additional constructs of a richer type system to an XML schema. As stated in

the description of Berger in the introductory portion of this response, neither Berger nor UIML disclose a one to one mapping as provided for in claim 12.

It is respectfully submitted that claim 12 is patentable over the cited art. Reconsideration and allowance is requested.

Claim 13

The examiner rejected claim 13 for the same rationale as the rejection of claim 1.

It is respectfully submitted that claim 13 is patentable over the cited art for the same reasons as set out for claim 1 above. Reconsideration and allowance is requested.

Claim 15

The examiner rejected claim 15 for the same rationale as the rejection of claim 1.

It is respectfully submitted that claim 15 is patentable over the cited art for the same reasons as set out for claim 1 above. Reconsideration and allowance is requested.

Claim 16

The examiner rejected claim 16 for the same rationale as the rejection of claim 1.

It is respectfully submitted that claim 16 is patentable over the cited art for the same reasons as set out for claim 1 above. Reconsideration and allowance is requested.

Claim 17 and 31

The examiner rejected claims 17 and 31 for the same rationale as the rejection of claim 2.

It is respectfully submitted that claims 17 and 31 are patentable over the cited art for the same reasons as set out for claim 2 above. Reconsideration and allowance is requested.

Claim 18 and 32

The examiner rejected claims 18 and 32 for the same rationale as the rejection of claim 3.

It is respectfully submitted that claims 18 and 32 are patentable over the cited art for the same reasons as set out for claim 3 above. Reconsideration and allowance is requested.

Claim 19 and 33

The examiner rejected claims 19 and 33 for the same rationale as the rejection of claim 4.

It is respectfully submitted that claims 19 and 33 are patentable over the cited art for the same reasons as set out for claim 4 above. Reconsideration and allowance is requested.

Claim 20 and 34

The examiner rejected claims 20 and 34 for the same rationale as the rejection of claim 5.

It is respectfully submitted that claims 20 and 34 are patentable over the cited art for the same reasons as set out for claim 5 above. Reconsideration and allowance is requested.

Claim 21 and 35

The examiner rejected claims 21 and 35 for the same rationale as the rejection of claim 6.

It is respectfully submitted that claims 21 and 35 are patentable over the cited art for the same reasons as set out for claim 5 above. Reconsideration and allowance is requested.

Claim 22 and 36

The examiner rejected claims 22 and 36 for the same rationale as the rejection of claim 7.

It is respectfully submitted that claims 22 and 36 are patentable over the cited art for the same reasons as set out for claim 7 above. Reconsideration and allowance is requested.

Claim 23 and 37

The examiner rejected claims 23 and 37 for the same rationale as the rejection of claim 8.

It is respectfully submitted that claims 23 and 37 are patentable over the cited art for the same reasons as set out for claim 8 above. Reconsideration and allowance is requested.

Claim 24 and 38

The examiner rejected claims 24 and 38 for the same rationale as the rejection of claim 9.

It is respectfully submitted that claims 24 and 38 are patentable over the cited art for the same reasons as set out for claim 9 above. Reconsideration and allowance is requested.

Claim 25 and 39

The examiner rejected claims 25 and 39 for the same rationale as the rejection of claim 10.

It is respectfully submitted that claims 25 and 39 are patentable over the cited art for the same reasons as set out for claim 10 above. Reconsideration and allowance is requested.

Claim 26 and 40

The examiner rejected claims 26 and 40 for the same rationale as the rejection of claim 11.

It is respectfully submitted that claims 26 and 40 are patentable over the cited art for the same reasons as set out for claim 11 above. Reconsideration and allowance is requested.

Claim 27 and 41

The examiner rejected claims 27 and 41 for the same rationale as the rejection of claim 12.

It is respectfully submitted that claims 27 and 41 are patentable over the cited art for the same reasons as set out for claim 12 above. Reconsideration and allowance is requested.

CONCLUSION

Applicants believe that the present Amendment is responsive to each of the points raised by the Examiner in the Final Office Action, and submit that Claims 1-13, 15-27 and 29-41 of the application are in condition for allowance. Favorable consideration and passage to issue of the application at the Examiner's earliest convenience is earnestly solicited.

Date: September 14, 2006

Eduardo M. Carreras
Registration No. 28,197

Woodcock Washburn LLP
One Liberty Place - 46th Floor
Philadelphia PA 19103
Telephone: (404) 459-5643
Facsimile: (404) 459-5734